Red Hat OpenStack Platform 17.0

Hyperconverged Infrastructure Guide

Understanding and configuring Hyperconverged Infrastructure on the Red Hat OpenStack Platform overcloud
Understanding and configuring Hyperconverged Infrastructure on the Red Hat OpenStack Platform

OpenStack Team
rhos-docs@redhat.com
Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation’s permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes the Red Hat OpenStack Platform implementation of hyperconvergence, which colocations Compute and Ceph Storage services on the same host.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ......................................................... 4

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................. 5

CHAPTER 1. CONFIGURING AND DEPLOYING A RED HAT OPENSTACK PLATFORM HYPERCONVERGED INFRASTRUCTURE .......................................................... 6
   1.1. HYPERCONVERGED INFRASTRUCTURE OVERVIEW .................................. 6

CHAPTER 2. DEPLOYING HCI HARDWARE ......................................................... 7
   2.1. CLEANING CEPH STORAGE NODE DISKS ................................................. 7
   2.2. REGISTERING NODES ........................................................................... 7
   2.3. VERIFYING AVAILABLE RED HAT CEPH STORAGE PACKAGES ............... 10
       2.3.1. Verifying cephadm package installation ......................................... 10
   2.4. DEPLOYING THE SOFTWARE IMAGE FOR AN HCI ENVIRONMENT .......... 10
   2.5. DESIGNATING NODES FOR HCI .......................................................... 11
   2.6. DEFINING THE ROOT DISK FOR MULTI-DISK CEPH CLUSTERS .............. 14
       2.6.1. Properties that identify the root disk ............................................. 15

CHAPTER 3. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER FOR HCI .... 16
   3.1. DEPLOYMENT PREREQUISITES .............................................................. 16
   3.2. THE OPENSTACK OVERCLOUD CEPH DEPLOY COMMAND ..................... 16
   3.3. CEPH CONFIGURATION OVERIDES FOR HCI ........................................ 16
   3.4. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER NAME .......... 17
   3.5. CONFIGURING NETWORK OPTIONS WITH THE NETWORK DATA FILE .... 18
   3.6. CONFIGURING NETWORK OPTIONS WITH A CONFIGURATION FILE ...... 19
   3.7. CONFIGURING A CRUSH HIERARCHY FOR AN OSD ............................... 20
   3.8. CONFIGURING CEPH SERVICE PLACEMENT OPTIONS .......................... 21
   3.9. CONFIGURING SSH USER OPTIONS FOR CEPH NODES ....................... 22
       3.9.1. Creating the SSH user before Red Hat Ceph Storage cluster creation 22
       3.9.2. Disabling the SSH user .................................................................. 23
   3.10. CONFIGURING THE CONTAINER REGISTRY .......................................... 23

CHAPTER 4. CUSTOMIZING THE RED HAT CEPH STORAGE CLUSTER FOR HCI .... 25
   4.1. CONFIGURATION OPTIONS ................................................................. 25
   4.2. GENERATING THE SERVICE SPECIFICATION (OPTIONAL) ....................... 26
   4.3. CEPH CONTAINERS FOR RED HAT OPENSTACK PLATFORM WITH RED HAT CEPH STORAGE ................................................................. 27
   4.4. CONFIGURING ADVANCED OSD SPECIFICATIONS ............................... 27
   4.5. MIGRATING FROM NODE-SPECIFIC OVERIDES .................................. 28
   4.6. ENABLING CEPH ON-WIRE ENCRYPTION ......................................... 28

CHAPTER 5. CUSTOMIZING THE STORAGE SERVICE FOR HCI .......................... 29
   5.1. CONFIGURING COMPUTE SERVICE RESOURCES FOR HCI .................... 29
   5.2. CONFIGURING A CUSTOM ENVIRONMENT FILE .................................. 29
   5.3. ENABLING CEPH METADATA SERVER ............................................... 30
   5.4. CEPH OBJECT GATEWAY OBJECT STORAGE ...................................... 31
   5.5. DEPLOYMENT OPTIONS FOR RED HAT OPENSTACK PLATFORM OBJECT STORAGE ................................................................. 31
       5.5.1. RHOSP deployment using Ceph Object Gateway (RGW) ................. 32
       5.5.2. RHOSP deployment using Swift instead of RGW ............................. 32
       5.5.3. RHOSP deployment without using object storage ............................ 32
   5.6. CONFIGURING THE BLOCK STORAGE BACKUP SERVICE TO USE CEPH 32
   5.7. CONFIGURING MULTIPLE BONDED INTERFACES FOR CEPH NODES .... 33
   5.8. INITIATING OVERCLOUD DEPLOYMENT FOR HCI ............................... 33
# Chapter 6. Verifying HCI Configuration

## 6.1. Verifying HCI Configuration

# Chapter 7. Scaling Hyperconverged Nodes

## 7.1. Scaling Up Hyperconverged Nodes in HCI Environments

## 7.2. Scaling Down Hyperconverged Nodes in HCI Environments

# Appendix A. Additional Information

## A.1. Guides and Resources for the Configuration of Your Hyperconverged Infrastructure Environment
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
We appreciate your input on our documentation. Tell us how we can make it better.

**Using the Direct Documentation Feedback (DDF) function**

Use the Add Feedback DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the *Multi-page HTML* format.
2. Ensure that you see the Feedback button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click Add Feedback.
5. Complete the Add Feedback field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click Submit.
CHAPTER 1. CONFIGURING AND DEPLOYING A RED HAT OPENSTACK PLATFORM HYPERCONVERGED INFRASTRUCTURE

1.1. HYPERCONVERGED INFRASTRUCTURE OVERVIEW

Red Hat OpenStack Platform (RHOSP) hyperconverged infrastructures (HCI) consist of hyperconverged nodes. In RHOSP HCI, the Compute and Storage services are colocated on these hyperconverged nodes for optimized resource use. You can deploy an overcloud with only hyperconverged nodes, or a mixture of hyperconverged nodes with normal Compute and Red Hat Ceph Storage nodes.

**NOTE**

You must use Red Hat Ceph Storage as the storage provider.

**TIP**

Use BlueStore as the back end for HCI deployments to make use of the BlueStore memory handling features.

Hyperconverged infrastructures are built using a variation of the deployment process described in Deploying Red Hat Ceph and OpenStack together with director. In this deployment scenario, RHOSP director deploys your cloud environment, which director calls the overcloud, and Red Hat Ceph Storage. You manage and scale the Ceph cluster itself separate from the overcloud configuration.
CHAPTER 2. DEPLOYING HCI HARDWARE

This section contains procedures and information about the preparation and configuration of hyperconverged nodes.

Prerequisites

- You have read Deploying an overcloud and Red Hat Ceph Storage in Deploying Red Hat Ceph and OpenStack together with director.

2.1. CLEANING CEPH STORAGE NODE DISKS

Ceph Storage OSDs and journal partitions require factory clean disks. This means the additional disks on Ceph Storage should be wiped by the Bare Metal Provisioning service (ironic) before installing the Ceph OSD services. You must delete all metadata from the disks or OSDs will not be created on them.

You can configure director to delete all disk metadata by default. With this option, the Bare Metal Provisioning service runs an additional step to boot the nodes and clean the disks each time the node is set to available. This process adds an additional power cycle after the first introspection and before each deployment. The Bare Metal Provisioning service uses the `wipefs --force --all` command to perform the clean.

Procedure

1. Add the following setting to your `/home/stack/undercloud.conf` file:

   ```
   clean_nodes=true
   ```

2. After you set this option, run the `openstack undercloud install` command to execute this configuration change.

   **WARNING**

   The `wipefs --force --all` command deletes all data and metadata on the disk, but it does not perform a secure erase. A secure erase takes much longer.

2.2. REGISTERING NODES

You must register the nodes to enable director to communicate with them.

Procedure

1. Create a node inventory file in JSON format in the `home` directory of the `stack` user:

   ```
   $ vi /home/stack/instackenv.json
   ```
2. Configure the inventory file with the hardware and power management details that director can use to register nodes:

```json
{
  "nodes": [
    {
      "mac": [
        "b1:b1:b1:b1:b1:b1"
      ],
      "cpu": "4",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "ipmi",
      "pm_user": "admin",
      "pm_password": "p@55w0rd!",
      "pm_addr": "192.0.2.205"
    },
    {
      "mac": [
        "b2:b2:b2:b2:b2:b2"
      ],
      "cpu": "4",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "ipmi",
      "pm_user": "admin",
      "pm_password": "p@55w0rd!",
      "pm_addr": "192.0.2.206"
    },
    {
      "mac": [
        "b3:b3:b3:b3:b3:b3"
      ],
      "cpu": "4",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "ipmi",
      "pm_user": "admin",
      "pm_password": "p@55w0rd!",
      "pm_addr": "192.0.2.207"
    },
    {
      "mac": [
        "c1:c1:c1:c1:c1:c1"
      ],
      "cpu": "4",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "ipmi",
      "pm_user": "admin",
      "pm_password": "p@55w0rd!",
      "pm_addr": "192.0.2.208"
    }
  ]
}``
3. Initialize the stack user, then import the `instackenv.json` inventory file into director:

```
$ source ~/stackrc
$ openstack overcloud node import ~/instackenv.json
```

The `openstack overcloud node import` command imports the inventory file and registers each node with director.

4. Assign the kernel and ramdisk images to each node:

```
$ openstack overcloud node configure <node>
```

The nodes are registered and configured in director.

### 2.3. VERIFYING AVAILABLE RED HAT CEPH STORAGE PACKAGES

To help avoid overcloud deployment failures, verify that the required packages exist on your servers.

#### 2.3.1. Verifying cephadm package installation

The `cephadm` package must be installed on at least one overcloud node to bootstrap the first node of the Ceph Storage cluster. The `cephadm` package is included in the `overcloud-hardened-uefi-full.qcow2` image. The `tripleo_cephadm` role uses the Ansible package module to ensure it is present in the image.

### 2.4. DEPLOYING THE SOFTWARE IMAGE FOR AN HCI ENVIRONMENT

Nodes configured for an HCI environment must use the `overcloud-hardened-uefi-full.qcow2` software image. Using this software image requires a Red Hat OpenStack Platform (RHOSP) subscription.

**Procedure**


2. Add or update the `image` property for nodes that require the `overcloud-hardened-uefi-full` image. You can set the image to be used on specific nodes, or for all nodes that use a specific role:

   **Specific nodes**
- name: Ceph
count: 3
instances:
- hostname: overcloud-ceph-0
  name: node00
  image:
    href: file:///var/lib/ironic/images/overcloud-minimal.qcow2
- hostname: overcloud-ceph-1
  name: node01
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
- hostname: overcloud-ceph-2
  name: node02
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2

All nodes configured for a specific role

- name: ComputeHCI
count: 3
defaults:
  image:
    href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
instances:
- hostname: overcloud-ceph-0
  name: node00
- hostname: overcloud-ceph-1
  name: node01
- hostname: overcloud-ceph-2
  name: node02

3. In the roles_data.yaml role definition file, set the rhsm_enforce parameter to False.

   rhsm_enforce: False

4. Run the provisioning command:

   (undercloud)$ openstack overcloud node provision \
   --stack overcloud \
   --output /home/stack/templates/overcloud-baremetal-deployed.yaml \
   /home/stack/templates/overcloud-baremetal-deploy.yaml

5. Pass the overcloud-baremetal-deployed.yaml environment file to the openstack overcloud deploy command.

2.5. DESIGNATING NODES FOR HCI

To designate nodes for HCI, you must create a new role file to configure the ComputeHCI role, and configure the bare metal nodes with a resource class for ComputeHCI.

Procedure

1. Log in to the undercloud as the stack user.
2. Source the `stackrc` credentials file:

   ```bash
   [stack@director ~]$ source ~/stackrc
   ```

3. Generate a new roles data file named `roles_data.yaml` that includes the `Controller` and `ComputeHCI` roles:

   ```bash
   (undercloud)$ openstack overcloud roles generate Controller ComputeHCI -o ~/roles.yaml
   ```

4. Open `roles_data.yaml` and ensure that it has the following parameters and sections:

<table>
<thead>
<tr>
<th>Section/Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role comment</td>
<td>Role: ComputeHCI</td>
</tr>
<tr>
<td>Role name</td>
<td>name: ComputeHCI</td>
</tr>
<tr>
<td>description</td>
<td>HCI role</td>
</tr>
<tr>
<td>HostnameFormatDefault</td>
<td>%stackname%-novaceph-%index%</td>
</tr>
<tr>
<td>deprecated_nic_config_name</td>
<td>ceph.yaml</td>
</tr>
</tbody>
</table>

5. Register the ComputeHCI nodes for the overcloud by adding them to your node definition template, `node.json` or `node.yaml`.

6. Inspect the node hardware:

   ```bash
   (undercloud)$ openstack overcloud node introspect --all-manageable --provide
   ```

7. Tag each bare metal node that you want to designate for HCI with a custom HCI resource class:

   ```bash
   (undercloud)$ openstack baremetal node set \       --resource-class baremetal.HCI <node>
   ```

   Replace `<node>` with the ID of the bare metal node.

8. Add the `ComputeHCI` role to your `/home/stack/templates/overcloud-baremetal-deploy.yaml` file, and define any predictive node placements, resource classes, or other attributes that you want to assign to your nodes:

   ```yaml
   - name: Controller
count: 3
   - name: ComputeHCI
count: 1
defaults:
    resource_class: baremetal.HCI
   ```

9. Open the `baremetal.yaml` file and ensure that it contains the network configuration necessary for HCI. The following is an example configuration:

   ```yaml
   - name: ComputeHCI
   ```
count: 3
hostname_format: compute-hci-%index%
defaults:
  profile: ComputeHCI
  network_config:
    template: /home/stack/templates/three-nics-vlans/compute-hci.j2
  networks:
  - network: ctlplane
    vif: true
  - network: external
    subnet: external_subnet
  - network: internalapi
    subnet: internal_api_subnet01
  - network: storage
    subnet: storage_subnet01
  - network: storage_mgmt
    subnet: storage_mgmt_subnet01
  - network: tenant
    subnet: tenant_subnet01

NOTE

Network configuration in the ComputeHCI role contains the storage_mgmt network. CephOSD nodes use this network to make redundant copies of data. The network configuration for the Compute role does not contain this network.

See Bare Metal Provisioning for more information.

10. Run the provisioning command:

```
(undercloud)$ openstack overcloud node provision \
  --stack overcloud \
  --output /home/stack/templates/overcloud-baremetal-deployed.yaml \ 
  /home/stack/templates/overcloud-baremetal-deploy.yaml
```

11. Monitor the provisioning progress in a separate terminal.

```
(undercloud)$ watch openstack baremetal node list
```

NOTE

The watch command renews every 2 seconds by default. The -n option sets the renewal timer to a different value.

12. To stop the watch process, enter Ctrl-c.

13. Verification: When provisioning is successful, the node state changes from available to active.

Additional resources

- For more information about registering nodes, see Registering nodes for the overcloud in the Director Installation and Usage guide.
For more information about inspecting node hardware, see Creating an inventory of the bare-metal node hardware in the Director Installation and Usage guide.

For more information on network configuration in the ComputeHCI role and the storage_mgmt network, see Bare Metal Provisioning.

2.6. DEFINING THE ROOT DISK FOR MULTI-DISK CEPH CLUSTERS

Most Ceph Storage nodes use multiple disks. When nodes use multiple disks, director must identify the root disk. By default, director writes the overcloud image to the root disk during the provisioning process.

Use this procedure to identify the root device by serial number. For more information about other properties you can use to identify the root disk, see Section 2.6.1, "Properties that identify the root disk".

Procedure

1. Verify the disk information from the hardware introspection of each node. The following command to displays the disk information of a node:

   ```bash
   (undercloud)$ openstack baremetal introspection data save 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 | jq ".inventory.disks"
   ```

   For example, the data for one node might show three disks:

   ```json
   [
     {
       "size": 299439751168,
       "rotational": true,
       "vendor": "DELL",
       "name": "/dev/sda",
       "wwn_vendor_extension": "0x1ea4d42c412a9632b",
       "wwn_with_extension": "0x61866da0f3807001ea4d42c412a9632b",
       "model": "PERC H330 Mini",
       "wwn": "0x61866da0f380700",
       "serial": "61866da0f3807001ea4d42c412a9632b"
     },
     {
       "size": 299439751168,
       "rotational": true,
       "vendor": "DELL",
       "name": "/dev/sdb",
       "wwn_vendor_extension": "0x1ea4e13c12e36ad6",
       "wwn_with_extension": "0x61866da0f380d001ea4e13c12e36ad6",
       "model": "PERC H330 Mini",
       "wwn": "0x61866da0f380d00",
       "serial": "61866da0f380d001ea4e13c12e36ad6"
     },
     {
       "size": 299439751168,
       "rotational": true,
       "vendor": "DELL",
       "name": "/dev/sdc",
       "wwn_vendor_extension": "0x1ea4e31e121cfb45",
       "wwn_with_extension": "0x61866da0f37f001ea4e31e121cfb45",
       "model": "PERC H330 Mini",
       "wwn": "0x61866da0f37f00",
       "serial": "61866da0f37f001ea4e31e121cfb45"
     }
   ]
   ```
2. On the undercloud, set the root disk for a node. Include the most appropriate hardware attribute value to define the root disk.

(undercloud)$ openstack baremetal node set --property root_device='{"serial": "<serial_number>"}' <node-uuid>

For example, to set the root device to disk 2, which has the serial number 61866da04f380d001ea4e13c12e36ad6, enter the following command:

(undercloud)$ openstack baremetal node set --property root_device='{"serial": "61866da04f380d001ea4e13c12e36ad6"}' 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0

**NOTE**

Configure the BIOS of each node to boot from the root disk that you choose. Configure the boot order to boot from the network first, then from the root disk.

Director identifies the specific disk to use as the root disk. When you run the `openstack overcloud node provision` command, director provisions and writes the overcloud image to the root disk.

### 2.6.1. Properties that identify the root disk

There are several properties that you can define to help director identify the root disk:

- **model** (String): Device identifier.
- **vendor** (String): Device vendor.
- **serial** (String): Disk serial number.
- **hctl** (String): Host:Channel:Target:Lun for SCSI.
- **size** (Integer): Size of the device in GB.
- **wwn** (String): Unique storage identifier.
- **wwn_with_extension** (String): Unique storage identifier with the vendor extension appended.
- **wwn_vendor_extension** (String): Unique vendor storage identifier.
- **rotational** (Boolean): True for a rotational device (HDD), otherwise false (SSD).
- **name** (String): The name of the device, for example: /dev/sdb1.

**IMPORTANT**

Use the **name** property only for devices with persistent names. Do not use **name** to set the root disk for any other devices because this value can change when the node boots.
CHAPTER 3. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER FOR HCI

This chapter describes how to configure and deploy the Red Hat Ceph Storage cluster for HCI environments.

3.1. DEPLOYMENT PREREQUISITES

Confirm the following has been performed before attempting to configure and deploy the Red Hat Ceph Storage cluster:

- Provision of bare metal instances and their networks using the Bare Metal Provisioning service (ironic). For more information about the provisioning of bare metal instances, see Bare Metal Provisioning.

3.2. THE OPENSTACK OVERCLOUD CEPH DEPLOY COMMAND

If you deploy the Ceph cluster using director, you must use the `openstack overcloud ceph deploy` command. For a complete listing of command options and parameters, see `openstack overcloud ceph deploy` in the Command Line Interface Reference.

The command `openstack overcloud ceph deploy --help` provides the current options and parameters available in your environment.

3.3. CEPH CONFIGURATION OVERRIDES FOR HCI

A standard format initialization file is an option for Ceph cluster configuration. This initialization file is then used to configure the Ceph cluster with either the `cephadm bootstrap --config <file_name>` or `openstack overcloud ceph deploy --config <file_name>` commands.

Colocating Ceph OSD and Compute services on hyperconverged nodes risks resource contention between Red Hat Ceph Storage and Compute services. This occurs because the services are not aware of the colocation. Resource contention can result in service degradation, which offsets the benefits of hyperconvergence.

Resource allocation can be tuned using an initialization file to manage resource contention. The following creates an initialization file called `initial-ceph.conf` and then uses the `openstack overcloud ceph deploy` command to configure the HCI deployment.

```bash
$ cat <<EOF > initial-ceph.conf
[osd]
  osd_memory_target_autotune = true
  osd_numa_auto_affinity = true

[mgr]
  mgr/cephadm/autotune_memory_target_ratio = 0.2
EOF
$ openstack overcloud ceph deploy --config initial-ceph.conf
```

The `osd_memory_target_autotune` option is set to `true` so that the OSD daemons adjust their memory consumption based on the `osd_memory_target config` option. The `autotune_memory_target_ratio` defaults to `0.7`. This indicates 70% of the total RAM in the system is the starting point from which any memory consumed by non-autotuned Ceph daemons are subtracted. Then the remaining memory is divided by the OSDs, assuming all OSDs have `osd_memory_target_autotune` set to `true`. For HCI
deployments, set the `mgr/cephadm/autotune_memory_target_ratio` to 0.2 to ensure more memory is available for the Compute service. The 0.2 value is a cautious starting point. After deployment, use the `ceph` command to change this value if necessary.

A two NUMA node system can host a latency sensitive Nova workload on one NUMA node and a Ceph OSD workload on the other NUMA node. To configure Ceph OSDs to use a specific NUMA node not used by the Compute workload, use either of the following Ceph OSD configurations:

- `osd_numa_node` sets affinity to a numa node
- `osd_numa_auto_affinity` automatically sets affinity to the NUMA node where storage and network match

If there are network interfaces on both NUMA nodes and the disk controllers are NUMA node 0, use a network interface on NUMA node 0 for the storage network and host the Ceph OSD workload on NUMA node 0. Host the Nova workload on NUMA node 1 and have it use the network interfaces on NUMA node 1. Setting `osd_numa_auto_affinity` to `true` to achieve this configuration. Alternatively, the `osd_numa_node` could be set directly to 0 and a value would not be set for `osd_numa_auto_affinity` so that it defaults to `false`.

When a hyperconverged cluster backfills as a result of an OSD going offline, the backfill process can be slowed down. In exchange for a slower recovery, the backfill activity has less of an impact on the collocated Compute workload. Red Hat Ceph Storage 5 has the following defaults to control the rate of backfill activity:

- `osd_recovery_op_priority = 3`
- `osd_max_backfills = 1`
- `osd_recovery_max_active_hdd = 3`
- `osd_recovery_max_active_ssd = 10`

**NOTE**

It is not necessary to pass these defaults in an initialization file as they are the default values. If values other than the defaults are desired for the initial configuration, add them to the initialization file with the required values before deployment. After deployment, use the command ‘ceph config set osd’.

### 3.4. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER NAME

You can deploy the Red Hat Ceph Storage cluster with a name that you configure. The default name is `ceph`.

**Procedure**

1. Log in to the undercloud node as the `stack` user.

2. Configure the name of the Ceph Storage cluster by using the following command:

   ```bash
   openstack overcloud ceph deploy --cluster <cluster_name>
   ```

   Alternatively:

   ```bash
   $ openstack overcloud ceph deploy --cluster central
   ```
NOTE

Keyring files are not created at this time. Keyring files are created during the overcloud deployment. Keyring files inherit the cluster name configured during this procedure. For more information about overcloud deployment see Section 5.8, "Initiating overcloud deployment for HCI”

In the example above, the Ceph cluster is named central. The configuration and keyring files for the central Ceph cluster would be created in /etc/ceph during the deployment process.

```
[root@oc0-controller-0 ~]# ls -l /etc/ceph/
total 16
-rw-------. 1 root root  63 Mar 26 21:49 central.client.admin.keyring
-rw-------. 1 root root 201 Mar 26 22:17 central.client.openstack.keyring
-rw-------. 1 root root 134 Mar 26 22:17 central.client.radosgw.keyring
-rw-r--r--. 1 root root 177 Mar 26 21:49 central.conf
```

Troubleshooting

The following error may be displayed if you configure a custom name for the Ceph Storage cluster:

`monclient: get_monmap_and_config cannot identify monitors to contact because`

If this error is displayed, use the following command after Ceph deployment:

```
cephadm shell --config <configuration_file> --keyring <keyring_file>
```

For example, if this error was displayed when you configured the cluster name to central, you would use the following command:

```
cephadm shell --config /etc/ceph/central.conf \ 
--keyring /etc/ceph/central.client.admin.keyring
```

The following command could also be used as an alternative:

```
cephadm shell --mount /etc/ceph:/etc/ceph
export CEPH_ARGS='--cluster central'
```

3.5. CONFIGURING NETWORK OPTIONS WITH THE NETWORK DATA FILE

The network data file describes the networks used by the Red Hat Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the stack user.

2. Create a YAML format file that defines the custom network attributes called network_data.yaml.
IMPORTANT

Using network isolation, the standard network deployment consists of two storage networks which map to the two Ceph networks:

- The storage network, `storage`, maps to the Ceph network, `public_network`. This network handles storage traffic such as the RBD traffic from the Compute nodes to the Ceph cluster.

- The storage network, `storage_mgmt`, maps to the Ceph network, `cluster_network`. This network handles storage management traffic such as data replication between Ceph OSDs.

3. Use the `openstack overcloud ceph deploy` command with the `--crush-hierarchy` option to deploy the configuration.

```
openstack overcloud ceph deploy \
    deployed_metal.yaml \
    -o deployed_ceph.yaml \
    --network-data network_data.yaml
```

IMPORTANT

The `openstack overcloud ceph deploy` command uses the network data file specified by the `--network-data` option to determine the networks to be used as the `public_network` and `cluster_network`. The command assumes these networks are named `storage` and `storage_mgmt` in network data file unless a different name is specified by the `--public-network-name` and `--cluster-network-name` options.

You must use the `--network-data` option when deploying with network isolation. The default undercloud (192.168.24.0/24) will be used for both the `public_network` and `cluster_network` if you do not use this option.

3.6. CONFIGURING NETWORK OPTIONS WITH A CONFIGURATION FILE

Network options can be specified with a configuration file as an alternative to the network data file.

IMPORTANT

Using this method to configure network options overwrites automatically generated values in `network_data.yaml`. Ensure you set all four values when using this network configuration method.

Procedure

1. Log in to the undercloud node as the `stack` user.

2. Create a standard format initialization file to configure the Ceph cluster. If you have already created a file to include other configuration options, you can add the network configuration to it.

3. Add the following parameters to the `[global]` section of the file:
   - `public_network`
- public_network
- cluster_network
- ms_bind_ipv4

**IMPORTANT**

Ensure the `public_network` and `cluster_network` map to the same networks as `storage` and `storage_mgmt`.

The following is an example of a configuration file entry for a network configuration with multiple subnets and custom networking names:

```
[global]
public_network = 172.16.14.0/24,172.16.15.0/24
cluster_network = 172.16.12.0/24,172.16.13.0/24
ms_bind_ipv4 = True
ms_bind_ipv6 = False
```

4. Use the command `openstack overcloud ceph deploy` with the `--config` option to deploy the configuration file.

```
$ openstack overcloud ceph deploy \
   --config initial-ceph.conf --network-data network_data.yaml
```

### 3.7. CONFIGURING A CRUSH HIERARCHY FOR AN OSD

You can configure a custom Controlled Replication Under Scalable Hashing (CRUSH) hierarchy during OSD deployment to add the OSD `location` attribute to the Ceph Storage cluster `hosts` specification. The `location` attribute configures where the OSD is placed within the CRUSH hierarchy.

**NOTE**

The `location` attribute sets only the initial CRUSH location. Subsequent changes of the attribute are ignored.

**Procedure**

1. Log in to the undercloud node as the `stack` user.

2. Source the `stackrc` undercloud credentials file:

   `$ source ~/stackrc`

3. Create a configuration file to define the custom CRUSH hierarchy, for example, `crush_hierarchy.yaml`.

4. Add the following configuration to the file:

   ```yaml
   ceph_crush_hierarchy:
   <osd_host>:
     root: default
     rack: <rack_num>
   <osd_host>:
   ```
root: default
rack: <rack_num>
<osd_host>:
root: default
rack: <rack_num>

- Replace `<osd_host>` with the hostnames of the nodes where the OSDs are deployed, for example, `ceph-0`.

- Replace `<rack_num>` with the number of the rack where the OSDs are deployed, for example, `r0`.

5. Deploy the Ceph cluster with your custom OSD layout:

```
openstack overcloud ceph deploy \
    deployed_metal.yaml \
    -o deployed_ceph.yaml \
    --osd-spec osd_spec.yaml \
    --crush-hierarchy crush_hierarchy.yaml
```

The Ceph cluster is created with the custom OSD layout.

The example file above would result in the following OSD layout.

```
ID  CLASS  WEIGHT     TYPE NAME                  STATUS  REWEIGHT  PRI-AFF
-1  0.02939  root default
-3  0.00980  rack r0
-2  0.00980  host ceph-node-00
  0  hdd  0.00980  osd.0    up   1.00000  1.00000
-5  0.00980  rack r1
-4  0.00980  host ceph-node-01
  1  hdd  0.00980  osd.1    up   1.00000  1.00000
-7  0.00980  rack r2
-6  0.00980  host ceph-node-02
  2  hdd  0.00980  osd.2    up   1.00000  1.00000
```

NOTE
Device classes are automatically detected by Ceph but CRUSH rules are associated with pools. Pools are still defined and created using the `CephCrushRules` parameter during the overcloud deployment.

Additional resources

3.8. CONFIGURING CEPH SERVICE PLACEMENT OPTIONS

You can define what nodes run what Ceph services using a custom roles file. A custom roles file is only necessary when default role assignments are not used because of the environment. For example, when deploying hyperconverged nodes, the predeployed compute nodes should be labeled as `osd` with a service type of `osd` to have a placement list containing a list of compute instances.
Service definitions in the roles_data.yaml file determine which bare metal instance runs which service. By default, the Controller role has the CephMon and CephMgr service while the CephStorage role has the CephOSD service. Unlike most composable services, Ceph services do not require heat output to determine how services are configured. The roles_data.yaml file always determines Ceph service placement even though the deployed Ceph process occurs before Heat runs.

Procedure

1. Log in to the undercloud node as the stack user.

2. Create a YAML format file that defines the custom roles.

3. Deploy the configuration file:

```
$ openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --roles-data custom_roles.yaml
```

3.9. CONFIGURING SSH USER OPTIONS FOR CEPH NODES

The openstack overcloud ceph deploy command creates the user and keys and distributes them to the hosts so it is not necessary to perform the procedures in this section. However, it is a supported option.

Cephadm connects to all managed remote Ceph nodes using SSH. The Red Hat Ceph Storage cluster deployment process creates an account and SSH key pair on all overcloud Ceph nodes. The key pair is then given to Cephadm so it can communicate with the nodes.

3.9.1. Creating the SSH user before Red Hat Ceph Storage cluster creation

You can create the SSH user before Ceph cluster creation with the openstack overcloud ceph user enable command.

Procedure

1. Log in to the undercloud node as the stack user.

2. Create the SSH user:

```
$ openstack overcloud ceph user enable
```

**NOTE**

The default user name is ceph-admin. To specify a different user name, use the -cephadm-ssh-user option to specify a different one.

```
openstack overcloud ceph user enable --cephadm-ssh-user <custom_user_name>
```

It is recommended to use the default name and not use the --cephadm-ssh-user parameter.

If the user is created in advance, use the parameter --skip-user-create when executing openstack overcloud ceph deploy.
3.9.2. Disabling the SSH user

Disabling the SSH user disables Cephadm. Disabling Cephadm removes the ability of the service to administer the Ceph cluster and prevents associated commands from working. It also prevents Ceph node overcloud scaling operations. It also removes all public and private SSH keys.

Procedure

1. Log in to the undercloud node as the stack user.

2. Use the command `openstack overcloud ceph user disable --fsid <FSID> ceph_spec.yaml` to disable the SSH user.

   **NOTE**
   
   The FSID is located in the `deployed_ceph.yaml` environment file.

   **IMPORTANT**
   
   The `openstack overcloud ceph user disable` command is not recommended unless it is necessary to disable Cephadm.

   **IMPORTANT**
   
   To enable the SSH user and Cephadm service after being disabled, use the `openstack overcloud ceph user enable --fsid <FSID> ceph_spec.yaml` command.

   **NOTE**
   
   This command requires the path to a Ceph specification file to determine:
   
   - Which hosts require the SSH user.
   - Which hosts have the `_admin` label and require the private SSH key.
   - Which hosts require the public SSH key.

   For more information about specification files and how to generate them, see Generating the service specification.

3.10. CONFIGURING THE CONTAINER REGISTRY

The undercloud can be configured to serve as a container registry for Ceph containers. By default, `openstack overcloud ceph deploy` pulls the Ceph container in the default `container_image_prepare_defaults.yaml` file. If the `push_destination` attribute is defined in the file, the overcloud is configured to access the local registry to download the Ceph container. This modifies the overcloud `/etc/hosts` and `/etc/containers/registries.conf` files unless the `--skip-hosts-config` and `--skip-container-registry-config` options are used or a `push_destination` is not defined.
NOTE

Red Hat Ceph version changes in each Red Hat OpenStack release. The current Ceph version can be viewed using this command:

```bash
$ egrep "ceph_namespace|ceph_image|ceph_tag" \
/usr/share/tripleo-common/container-images/container_image_prepare_defaults.yaml
```

Procedure

1. Log in to the undercloud node as the stack user.

2. Create a YAML format file that defines the custom container registry. The following is an example of a custom container registry file called `custom_container_image_prepare.yaml`.

   ```yaml
   ContainerImageRegistryCredentials:
   quay.io/ceph-ci:
   quay_username: quay_password
   ```

3. Use the `openstack overcloud ceph deploy` command with the `--container-image-prepare` option to define the custom container registry.

   ```bash
   openstack overcloud ceph deploy \
   deployed_metal.yaml \
   -o deployed_ceph.yaml \
   --container-image-prepare custom_container_image_prepare.yaml
   ```
CHAPTER 4. CUSTOMIZING THE RED HAT CEPH STORAGE CLUSTER FOR HCI

Red Hat OpenStack Platform (RHOSP) director uses a default configuration to deploy containerized Red Hat Ceph Storage. You can customize Ceph Storage by overriding the default settings.

Prerequisites

- The servers are deployed and their storage networks configured.
- The deployed bare metal file as output by `openstack overcloud node provision -o ~/deployed_metal.yaml` ...

4.1. CONFIGURATION OPTIONS

Red Hat Ceph Storage provides several options for customizing the initial configuration of the Ceph cluster.

Initial Red Hat Ceph Storage configuration can be applied using a standard format initialization (ini) file with a .conf extension.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Optional: Use a standard format initialization (ini) file to configure the Ceph cluster.
   - a. Create the file with configuration options. The following is an example of a simple configuration file.

```ini
[globals]
osd crush chooseleaf type = 0
log_file = /var/log/ceph/$cluster-$type.$id.log

[mon]
mon_cluster_log_to_syslog = true
```

   - b. Use the `openstack overcloud ceph deploy --config <configuration_file_name>` command to deploy the configuration.

```bash
$ openstack overcloud ceph deploy --config initial-ceph.conf
```
IMPORTANT

The `deployed_ceph.yaml` environment file output by the `openstack overcloud ceph deploy` command has the `ApplyCephConfigOverridesOnUpdate` attribute set to `true`. This value allows services not covered by during the initial cluster creation, such as RGW, to be set during the overcloud deployment. After the overcloud deployment is complete, update the `deployed_ceph.yaml` file, or the file used for a similar purpose in your environment, to set the `ApplyCephConfigOverridesOnUpdate` to `false`. Subsequent Ceph configuration changes made during the operation of your environment should be made with the `ceph config` command. For more information on the `ApplyCephConfigOverridesOnUpdate` and `CephConfigOverrides` parameters see Overcloud Parameters.

3. Optional: Use the command `openstack overcloud ceph deploy --single-host-defaults` to configure the Ceph cluster to run on a single instance.

NOTE

This option is only available for testing environments. It is not supported for production environments.

4. Optional: Use the command `openstack overcloud ceph deploy --force \ --cephadm-extra-args <optional_arguments> \` to pass any additional configuration values to the `cephadm bootstrap` command. If the arguments `--log-to-file` and `--skip-prepare-host` are required, the command `openstack overcloud ceph deploy --force \ --cephadm-extra-args '--log-to-file --skip-prepare-host' \` would be used. The underlying `cephadm bootstrap` command `cephadm bootstrap --log-to-file --skip-prepare-host` would be executed.

NOTE

The `--force option` is required when using `--cephadm-extra-args` because not all possible options ensure a functional deployment.

4.2. GENERATING THE SERVICE SPECIFICATION (OPTIONAL)

The Red Hat Ceph Storage cluster service specification is a YAML file that describes the deployment of Ceph services. It is automatically generated by `tripleo` before the Ceph cluster is deployed and does not typically have to be generated separately.

A custom service specification can be created to customize the Red Hat Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the `stack` user.

2. Use the command `openstack overcloud ceph spec` command to generate the specification file. The following example would output the specification file at location specified by the `-o` switch.

   ```
   openstack overcloud ceph spec -o '~/ceph_spec.yaml'
   ```
3. Edit the generated file with the required configuration.

4. Use the command `openstack overcloud ceph deploy` command to deploy the custom service specification.

```
openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --ceph-spec ~/ceph_spec.yaml
```

4.3. CEPH CONTAINERS FOR RED HAT OPENSTACK PLATFORM WITH RED HAT CEPH STORAGE

To configure Red Hat Openstack Platform (RHOSP) to use Red Hat Ceph Storage with NFS Ganesha, you must have a Ceph container. You do not require a Ceph container if the external Ceph Storage cluster is only going to provide Block (through RBD) or Object (through RGW) storage.

To be compatible with Red Hat Enterprise Linux 9, RHOSP 17 requires Red Hat Ceph Storage 5 (Ceph package 16.x). The Ceph Storage 5 container is hosted at `registry.redhat.io`, a registry that requires authentication. For more information, see Container image preparation parameters.

4.4. CONFIGURING ADVANCED OSD SPECIFICATIONS

All disks, except for the one where the operating system is installed, are used as OSDs by default. This is because the default OSD specification file has the following definition.

```
data_devices:
  all: true
```

The `data_devices` attribute is one of several OSD-related attributes used to configure OSDs in the Ceph service specification. See OSD Service Specification lists the attributes available for OSD configuration.

**Procedure**

1. Log in to the undercloud node as the `stack` user.

2. Create a YAML format file called that defines the OSD specification. The following is an example of a custom OSD specification.

```
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

This example would create an OSD specification where all rotating devices will be data devices and all non-rotating devices will be used as shared devices. This is because when the dynamic Ceph service specification is built, whatever is in the specification file is appended to the section of the specification if the service_type is `osd`.

3. Use the command `openstack overcloud ceph deploy \ --osd-spec <osd_specification_file>` to deploy the configuration.

```
$ openstack overcloud ceph deploy \ --osd_spec osd_spec.yaml \
```
The following is another example of an `osd_spec.yaml` file:

```yaml
data_devices:
  model: 'SAMSUNG'
  osds_per_device: 2
```

In this example, the `model` parameter is used with `data_devices` to only create OSDs on that model of device, with two OSDs per device. The `osds_per_device` parameter is not under `data_devices`.

### 4.5. MIGRATING FROM NODE-SPECIFIC OVERRIDES

Prior to Red Hat OpenStack Platform 17.0, node-specific overrides were used to manage non-homogenous server hardware. This is now accomplished using a custom OSD specification file. See Configuring advanced OSD specifications for information on how to create a custom OSD specification file.

### 4.6. ENABLING CEPH ON-WIRE ENCRYPTION

You can enable encryption for all Ceph traffic over the network with the introduction of the messenger version 2 protocol. The `secure mode` setting for messenger v2 encrypts communication between Ceph daemons and Ceph clients, giving you end-to-end encryption.

**Procedure**

1. Configure the initial `ceph.conf` with the directives described in Enabling the messenger v2 protocol in the Red Hat Ceph Storage Data Security and Hardening Guide.

For more information about Ceph on-wire encryption, see Ceph on-wire encryption in the Red Hat Ceph Storage Architecture Guide.
CHAPTER 5. CUSTOMIZING THE STORAGE SERVICE FOR HCI

Red Hat OpenStack Platform (RHOSP) director provides the necessary heat templates and environment files to enable a basic Ceph Storage configuration.

Director uses the `/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml` environment file to add additional configuration to the Ceph cluster deployed by `openstack overcloud ceph deploy`.

For more information about containerized services in RHOSP, see Configuring a basic overcloud with the CLI tools in Director Installation and Usage.

5.1. CONFIGURING COMPUTE SERVICE RESOURCES FOR HCI

Colocating Ceph OSD and Compute services on hyperconverged nodes risks resource contention between Red Hat Ceph Storage and Compute services. This occurs because the services are not aware of the collocation. Resource contention can result in service degradation, which offsets the benefits of hyperconvergence.

Configuring the resources used by the Compute service mitigates resource contention and improves HCI performance.

Procedure

1. Log in to the undercloud host as the stack user.

2. Source the stackrc undercloud credentials file:

   ```bash
   $ source ~/stackrc
   ```

3. Add the `NovaReservedHostMemory` parameter to the `ceph-overrides.yaml` file. The following is a usage example.

   ```yaml
   parameter_defaults:
   ComputeHCIParameters:
   NovaReservedHostMemory: 75000
   ```

   The `NovaReservedHostMemory` parameter overrides the default value of `reserved_host_memory_mb` in `/etc/nova/nova.conf`. This parameter is set to stop Nova scheduler giving memory, that a Ceph OSD needs, to a virtual machine.

   The example above reserves 5 GB per OSD for 10 OSDs per host in addition to the default reserved memory for the hypervisor. In an IOPS-optimized cluster, you can improve performance by reserving more memory per OSD. The 5 GB number is provided as a starting point that you can further refine as necessary.

   **IMPORTANT**

   Include this file when you use the `openstack overcloud deploy` command.

5.2. CONFIGURIGN A CUSTOM ENVIRONMENT FILE

Director applies basic, default settings to the deployed Red Hat Ceph Storage cluster. You must define any additional configurations in a custom environment file.
Procedure

1. Create the file `storage-config.yaml` in `/home/stack/templates/`. In this example, the `~/templates/storage-config.yaml` file contains most of the overcloud-related custom settings for your environment. Parameters that you include in the custom environment file override the corresponding default settings from the `/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml` file.

2. Add a `parameter_defaults` section to `~/templates/storage-config.yaml`. This section contains custom settings for your overcloud.

3. Optional: Set the following options under `parameter_defaults` according to your requirements:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CinderEnableIscsiBackend</td>
<td>Enables the iSCSI backend</td>
<td>false</td>
</tr>
<tr>
<td>CinderEnableRbdBackend</td>
<td>Enables the Ceph Storage backend</td>
<td>true</td>
</tr>
<tr>
<td>CinderBackupBackend</td>
<td>Sets ceph or swift as the backend</td>
<td>ceph</td>
</tr>
<tr>
<td></td>
<td>for volume backups.</td>
<td></td>
</tr>
<tr>
<td>NovaEnableRbdBackend</td>
<td>Enables Ceph Storage for Nova</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>ephemeral storage</td>
<td></td>
</tr>
<tr>
<td>GlanceBackend</td>
<td>Defines which back end the Image</td>
<td>rbd</td>
</tr>
<tr>
<td></td>
<td>service should use: rbd (Ceph),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>swift, or file</td>
<td></td>
</tr>
</tbody>
</table>

NOTE

You can omit an option from `~/templates/storage-config.yaml` if you intend to use the default settings.

The contents of your custom environment file change depending on the settings that you apply in the following sections. See for a completed example.

5.3. ENABLING CEPH METADATA SERVER

The Ceph Metadata Server (MDS) runs the `ceph-mds` daemon, which manages metadata related to files stored on CephFS. CephFS can be consumed through NFS. For more information about using CephFS through NFS, see File System Guide and Deploying the Shared File Systems service with CephFS through NFS.

NOTE

Red Hat supports deploying Ceph MDS only with the CephFS through NFS back end for the Shared File Systems service.

Procedure
- To enable the Ceph Metadata Server, invoke the following environment file when you create your overcloud: `/usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml`

**NOTE**

By default, the Ceph Metadata Server is deployed on the Controller node. You can deploy the Ceph Metadata Server on its own dedicated node.

### 5.4. CEPH OBJECT GATEWAY OBJECT STORAGE

The Ceph Object Gateway (RGW) provides applications with an interface to access object storage capabilities within a Red Hat Ceph Storage cluster.

When you use director to deploy Ceph, director automatically enables RGW, which is a direct replacement for the Object Storage service (swift). All other services that normally use Swift can start using RGW instead without additional configuration. Swift remains available as an object storage option for upgraded Ceph clusters.

Now that RGW is enabled by default, there is no requirement for a separate RGW environment file to enable it. For more information about environment files that you can pass to the `openstack overcloud deploy` command for other object storage options, see Section 5.5, “Deployment options for Red Hat OpenStack Platform object storage”.

By default, Ceph Storage allows 250 placement groups per Object Storage Daemon (OSD). When you enable RGW, Ceph Storage creates the following six additional pools that are required by RGW:

- `.rgw.root`
- `default.rgw.control`
- `default.rgw.meta`
- `default.rgw.log`
- `default.rgw.buckets.index`
- `default.rgw.buckets.data`

**NOTE**

In your deployment, `default` is replaced with the name of the zone to which the pools belong.

Additional resources

- For more information about RGW, see the Red Hat Ceph Storage Object Gateway Guide.
- For more information about using RGW instead of Swift, see the Block Storage Backup Guide.

### 5.5. DEPLOYMENT OPTIONS FOR RED HAT OPENSTACK PLATFORM OBJECT STORAGE

When you have deployed a Red Hat Ceph Storage cluster using the `openstack overcloud ceph deploy`
command, you deploy the overcloud using the `openstack overcloud deploy` command. When you have deployed the overcloud, you can pass different environment files to the `openstack overcloud deploy` command for different object storage options.

5.5.1. RHOSP deployment using Ceph Object Gateway (RGW)

To deploy RGW as described in Section 5.4, “Ceph Object Gateway object storage”, include the following environment file:

```
-e environments/cephadm/cephadm.yaml
```

This environment configures both Ceph block storage (RBD) and RGW.

5.5.2. RHOSP deployment using Swift instead of RGW

To deploy Ceph with Swift object storage instead of RGW, include the following environment file:

```
-e environments/cephadm/cephadm-rbd-only.yaml
```

The `cephadm-rbd-only.yaml` file configures Ceph RBD but not RGW. If you use this file, object storage is still installed by default using Swift.

**NOTE**

If you used Swift object storage before upgrading your Red Hat Ceph Storage cluster, you can continue to use Swift instead of RGW by replacing the `environments/ceph-ansible/ceph-ansible.yaml` file with the `environments/cephadm/cephadm-rbd-only.yaml` during the upgrade. For more information, see the upgrade documentation. RHOSP does not support migration from Swift to RGW.

5.5.3. RHOSP deployment without using object storage

To deploy Ceph with RBD but not with RGW or Swift, include the following environment files:

```
-e environments/cephadm/cephadm-rbd-only.yaml
-e environments/disable-swift.yaml
```

The `cephadm-rbd-only.yaml` file configures RBD but not RGW. The `disable-swift.yaml` file ensures that the Object Storage service (swift) does not deploy.

5.6. CONFIGURING THE BLOCK STORAGE BACKUP SERVICE TO USE CEPH

The Block Storage Backup service (`cinder-backup`) is disabled by default. To enable the Block Storage Backup service, complete the following steps:

**Procedure**

- Invoke the following environment file when you create your overcloud: `/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml`. 

5.7. CONFIGURING MULTIPLE BONDED INTERFACES FOR CEPH NODES

Use a bonded interface to combine multiple NICs and add redundancy to a network connection. If you have enough NICs on your Ceph nodes, you can create multiple bonded interfaces on each node to expand redundancy capability.

You can then use a bonded interface for each network connection that the node requires. This provides both redundancy and a dedicated connection for each network.

See Provisioning the overcloud networks in the Director Installation and Usage guide for information and procedures.

5.8. INITIATING OVERCLOUD DEPLOYMENT FOR HCI

To implement the changes you made to your Red Hat OpenStack Platform (RHOSP) environment, you must deploy the overcloud.

Prerequisites

- Before undercloud installation, set `generate_service_certificate=false` in the `undercloud.conf` file. Otherwise, you must configure SSL/TLS on the overcloud as described in Enabling SSL/TLS on overcloud public endpoints in the Security and Hardening Guide.

**NOTE**

If you want to add Ceph Dashboard during your overcloud deployment, see Adding the Red Hat Ceph Storage Dashboard to an overcloud deployment in Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director.

Procedure

- Deploy the overcloud. The deployment command requires additional arguments, for example:

```
$ openstack overcloud deploy --templates -r /home/stack/templates/roles_data_custom.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml
  -e /home/stack/templates/storage-config.yaml
  -e /home/stack/templates/deployed-ceph.yaml
  --ntp-server pool.ntp.org
```

The example command uses the following options:

- **--templates** - Creates the overcloud from the default heat template collection, `/usr/share/openstack-tripleo-heat-templates/`.

- **-r /home/stack/templates/roles_data_custom.yaml** - Specifies a customized roles definition file.

- **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml** - Sets the director to finalize the previously deployed Ceph Storage cluster. This environment file deploys RGW by default. It also creates pools, keys, and daemons.

- `e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml` - Enables the Block Storage Backup service.

- `e /home/stack/templates/storage-config.yaml` - Adds the environment file that contains your custom Ceph Storage configuration.

- `e /home/stack/templates/deployed-ceph.yaml` - Adds the environment file that contains your Ceph cluster settings, as output by the `openstack overcloud ceph deploy` command run earlier.

- `--ntp-server pool.ntp.org` - Sets the NTP server.

**NOTE**

For a full list of options, run the `openstack help overcloud deploy` command.

### Additional resources

- For more information, see Configuring a basic overcloud with the CLI tools in the Director Installation and Usage guide.
CHAPTER 6. VERIFYING HCI CONFIGURATION

After deployment is complete, verify the HCI environment is properly configured.

6.1. VERIFYING HCI CONFIGURATION

After the deployment of the HCI environment, verify that the deployment was successful with the configuration specified.

Procedure

1. Start a ceph shell.

2. Confirm NUMA and memory target configuration:

   ```
   [ceph: root@oc0-controller-0 ]$ ceph config dump | grep numa
   osd                  advanced  osd_numa_auto_affinity             true
   [ceph: root@oc0-controller-0 ]$ ceph config dump | grep autotune
   osd                  advanced  osd_memory_target_autotune           true
   [ceph: root@oc0-controller-0 ]$ ceph config get mgr
   mgr/cephadm/autotune_memory_target_ratio
   0.200000
   ```

3. Confirm specific OSD configuration:

   ```
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_memory_target
   4294967296
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_memory_target_autotune
   true
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_numa_auto_affinity
   true
   ```

4. Confirm specific OSD backfill configuration:

   ```
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_recovery_op_priority
   3
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_max_backfills
   1
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_recovery_max_active_hdd
   3
   [ceph: root@oc0-controller-0 ]$ ceph config get osd.11 osd_recovery_max_active_ssd
   10
   ```

5. Confirm the `reserved_host_memory_mb` configuration on the Compute node.

   ```
   $ sudo podman exec -ti nova_compute /bin/bash
   bash-5.1$ grep reserved_host_memory_mb /etc/nova/nova.conf
   ```
CHAPTER 7. SCALING HYPERCONVERGED NODES

To scale HCI nodes up or down, the same principles and methods for scaling Compute nodes or Red Hat Ceph Storage nodes apply.

7.1. SCALING UP HYPERCONVERGED NODES IN HCI ENVIRONMENTS

To scale up hyperconverged nodes in HCI environments follow the same procedure for scaling up non-hyperconverged nodes. For more information, see Adding nodes to the overcloud.

NOTE

When you tag new nodes, remember to use the right flavor.

7.2. SCALING DOWN HYPERCONVERGED NODES IN HCI ENVIRONMENTS

To scale down hyperconverged nodes in HCI environments you must rebalance the Ceph OSD services on the HCI node, migrate instances from the HCI nodes, and remove the Compute nodes from the overcloud.

Procedure

1. Disable and rebalance the Ceph OSD services on the HCI node. This step is necessary because director does not automatically rebalance the Red Hat Ceph Storage cluster when you remove HCI or Red Hat Ceph Storage nodes. See Scaling the Ceph Storage cluster in Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director for more information.

2. Migrate the instances from the HCI nodes. For more information, see Migrating virtual machines between Compute nodes in the Configuring the Compute Service for Instance Creation guide.

3. Remove the Compute nodes from the overcloud. For more information, see Removing Compute nodes.
A.1. GUIDES AND RESOURCES FOR THE CONFIGURATION OF YOUR HYPERCONVERGED INFRASTRUCTURE ENVIRONMENT

The following guides contain additional information and procedures that can aid in the configuration of your hyperconverged infrastructure environment.

- **Deploying Red Hat Ceph and OpenStack together with director**
  - This guide provides information about using the Red Hat OpenStack Platform director to create an overcloud with a Red Hat Ceph Storage cluster. This includes instructions for customizing your Ceph cluster through the director.

- **Director Installation and Usage**
  - This guide provides guidance on the end-to-end deployment of a Red Hat OpenStack Platform environment. This includes installing the director, planning your environment, and creating an OpenStack environment with the director.

- **Networking Guide**
  - This guide provides details on Red Hat OpenStack Platform networking tasks.

- **Storage Guide**
  - This guide details the different procedures for using and managing persistent storage in a Red Hat OpenStack Platform environment. It also includes procedures for configuring and managing the respective OpenStack service of each persistent storage type.

- **Bare Metal Provisioning**
  - This guide provides details on the installation and configuration of the Bare Metal Provisioning service in the overcloud of a Red Hat OpenStack Platform environment to provision and manage physical machines for cloud users.

- **Security and Hardening Guide**
  - This guide provides good practice advice and conceptual information about hardening the security of a Red Hat OpenStack Platform environment.

- **Release Notes**
  - This document outlines the major features, enhancements, and known issues in this release of Red Hat OpenStack Platform.